Days to Pay Database

Creating the Days to Pay Reports

General Information:

As part of the Commonwealth's ongoing effort to ensure timely payment of its bills, the Office of the State Comptroller has created a database system that monitors the amount of time it takes for transactions to be paid by the various departments. The Commonwealth's Bill Paying Policy[1] requires that all bills be paid within 30 days of the receipt of an invoice or the completion of services (whichever comes later) *unless* the contract in question provides for a different payment schedule. For our purposes, the number of days that a bill takes to be paid (henceforth referred to as "**Days to Pay**" or "**Actual Days to Pay**") is the date entered on the PV subtracted from the date that the State Treasurer's Office mailed the check.

The Comptroller's Office uses a series of reports for the purpose of monitoring the Days to Pay that provide statistics regarding how many (and what percentage) of a department's transactions are paid under 30 days, between 30 and 45 days, and over 45 days. Individual departments—for more immediate monitoring—can duplicate these reports using similar methods, as outlined below.

Requirements:

- Microsoft Access 2.0 or higher
- Information Warehouse Access (Including: Object Table, Expenditure Detail Table and Warrant Ledger 2000 Table)
- Working knowledge of MS Access querying and reporting techniques
- Microsoft Excel 5.0 or higher (for analysis of data outside scope of these reports)

Key Terms:

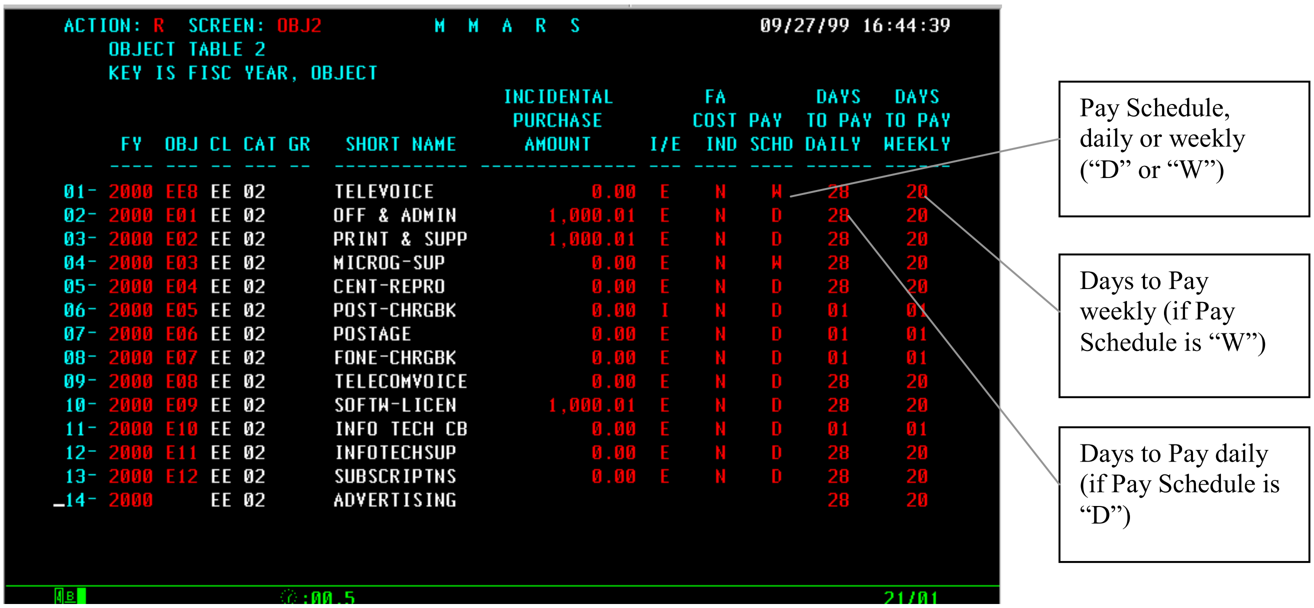
- **Date Of Record—**The PV Invoice Receipt Date/Voucher Date, as entered on the PV. This date is (by Commonwealth Policy) the date time-stamped on an invoice upon entering Commonwealth property or the date that goods are delivered or services are completed (the later of the two).
- **Payment Issue Date—**The date in which the State Treasurer's Office delivers a check to the Post Office or transmits the issuance file of an EFT to a bank
- **Days to Pay—**The number of days a bill takes to be officially paid according to principles outlined in the Commonwealth's Bill Paying Policy. It is equivalent to **Payment Issue Date** minus **Date of Record.**
- **20 Day Weekly Object Code—**Any object code which (according to **OBJ2** table in MMARS or **dbo_Object** in the Warehouse) causes a PV to be automatically placed on the *Weekly Warrant* 20 days after the **Date of Record**
- **28 Day Daily Object Code—**Any object code which (according to **OBJ2** table in MMARS or **dbo_Object** in the Warehouse) causes a PV to be automatically placed on the *Daily Warrant* 28 days after the **Date of Record**
- **OBJ2 Table—**The table in MMARS containing information regarding Days to Pay (see following section for a description of this table)
- **dbo_Object—**The Information Warehouse equivalent of OBJ2; allows user to query for 20 Day Weekly or 28 Day Daily object codes (see following section for information regarding the setup of these queries)

[1] See MMARS Memo #289 for more details regarding this policy

Examples of OBJ2 Table and dbo_Object Table

The Comptroller's Office evaluates transactions containing 20 Day Weekly object codes differently from those containing 28 Days Daily object codes. To determine which object codes are 20 Day Weekly and which are 20 Day Daily, it is necessary to view either the **OBJ2** table in MMARS or the Warehouse table **dbo_Object** (the latter is recommended, since the data may be used in MS Access). The relevant fields containing this information are shown below.

OBJ2 table (MMARS)


ACTION: R SCREEN: OBJ2 M M A R S 09/27/99 16:44:39
OBJECT TABLE 2
KEY IS FISC YEAR, OBJECT

	FY	OBJ	CL	CAT	GR	SHORT NAME	INCIDENTAL PURCHASE AMOUNT	I/E	FA COST IND	PAY SCHD	DAYS TO PAY DAILY	DAYS TO PAY WEEKLY
01-	2000	EE8	EE	02		TELEVOICE	0.00	E	N	W	28	20
02-	2000	E01	EE	02		OFF & ADMIN	1,000.01	E	N	D	28	20
03-	2000	E02	EE	02		PRINT & SUPP	1,000.01	E	N	D	28	20
04-	2000	E03	EE	02		MICROG-SUP	0.00	E	N	W	28	20
05-	2000	E04	EE	02		CENT-REPRO	0.00	E	N	D	28	20
06-	2000	E05	EE	02		POST-CHRGBK	0.00	I	N	D	01	01
07-	2000	E06	EE	02		POSTAGE	0.00	E	N	D	01	01
08-	2000	E07	EE	02		FONE-CHRGBK	0.00	E	N	D	01	01
09-	2000	E08	EE	02		TELECOMVOICE	0.00	E	N	D	28	20
10-	2000	E09	EE	02		SOFTW-LICEN	1,000.01	E	N	D	28	20
11-	2000	E10	EE	02		INFO TECH CB	0.00	E	N	D	01	01
12-	2000	E11	EE	02		INFOTECHSUP	0.00	E	N	D	28	20
13-	2000	E12	EE	02		SUBSCRIPTNS	0.00	E	N	D	28	20
14-	2000		EE	02		ADVERTISING					28	20

:00.5 21/01

dbo_Object table (Information Warehouse)

Note: If these queries are saved, it is possible to link them to the 20 Day Weekly and 28 Day Daily queries described in Sections 2 and 3. This method saves considerable data entry in creating those queries.

I. Finding 20 Days Weekly objects

Field Name	*Tbl*	*Criteria*	*Description*
Fiscal_Year	1	"2000"	Selects fiscal year for object codes
Object	1		
Object_Name	1		
Object_Class	1		
Payment_Schedule	1	"W"	Selects Weekly object codes
Days_To_Pay_Daily	1		
Days_To_Pay_Weekly	1	20	Selects 20 Days to Pay object codes

II. Finding 28 Days Daily objects

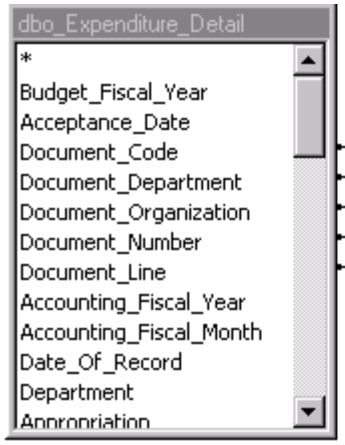
Field Name	*Tbl*	*Criteria*	*Description*
Fiscal_Year	1	"2000"	Selects fiscal year for object codes
Object	1		
Object_Name	1		
Object_Class	1		
Payment_Schedule	1	"D"	Selects Daily object codes
Days_To_Pay_Daily	1	28	Selects 28 Days to Pay object codes
Days_To_Pay_Weekly	1		

1. Building the main PV and RP database table (Make-Table Query)

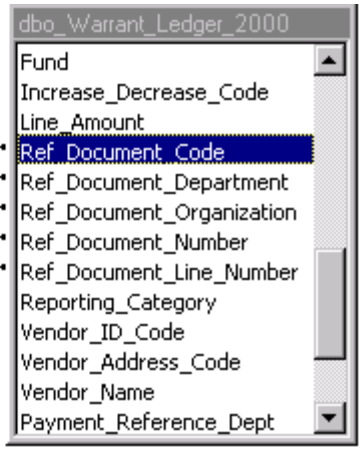
The first step in building the Days to Pay Database is creating a table which contains all the PVs and RPs whose payments were made during the desired quarter. In this first query, the Warehouse tables **dbo_Expenditure_Detail** (1) and **dbo_Warrant_Ledger_2000** (2) are used to acquire the appropriate data (see representations below).

It is also during this first step that the number of days a document took to pay is recorded (**Actual Days to Pay** field). The days to pay are calculated by means of subtracting the **Date_of_Record** field (the date that an invoice was received by the department) from the **Payment_Issue_Date** field (the date that the check was sent out by the State Treasurer). This field is created by entering the following expression in the field name area of the query design: **Actual Days to Pay: [Payment_Issue_Date]-[Date_Of_Record]**
The expression must be entered *exactly*, or the query will not run.

Attached Warehouse Table 1



Attached Warehouse Table 2



Field Name	*Tbl*	*Criteria*	*Description*
Budget_Fiscal_Year	1	"2000"	Selects fiscal year for transactions
Document_Code	1	"PV" Or "RP"	Selects PV or RP transactions
Document_Department	1		
Document_Organization	1		
Document_Number	1		
Document_Line	1		
Object	1		
Line_Amount	1		
Date_of_Record	1		
Payment_Issue_Date	2	Between #m/d/yy# And #m/d/yy#	Selects Payment Dates for desired period
Actual Days to Pay: [Payment_Issue_Date]-[Date_of_Record]	-		Creates **Actual Days to Pay** field by subtracting **Date_of_Record** from **Payment_Issue_Date** (see explanation above)
Debit_Credit_Code	2	"d"	Selects debits only
Warrant_Number	2		

Output Table: 4th quarter data main database

Description of Required Fields

- **Budget_Fiscal_Year:** Fiscal Year in which the document was budgeted
- **Document_Code:** Two letter MMARS transaction code from Document ID; the relevant document codes are RP (Ready Payment) and PV (Payment Voucher)
- **Document_Department:** Field is auto-filled with your department ID
- **Document_Number:** Seven digit Document ID number
- **Document_Line:** Line number of transaction within a document

- **Object:** Object code of transaction (see MMARS table **OBJ2** for descriptions of individual object codes and relevant Days to Pay information)
- **Line_Amount:** Dollar value of the document line
- **Date_of_Record:** PV Date/Voucher Date as entered on the PV in MMARS; this date, by Commonwealth Bill Paying Policy, is either the date stamped on a received invoice, or that date that a service was completed, *whichever is later*
- **Payment_Issue_Date:** Date a check/EFT was sent out by the State Treasurer's Office (invoice officially paid)
- **Debit_Credit_Code:** One letter code indicating whether a transaction is a debit or credit; only debits ("d") are considered by this database
- **Warrant_Number:** (optional) The warrant number on which a payment was made

2. Creating the 20 Day Weekly Main Table

This table contains all the PVs and RPs from the main table that have 20 Day Weekly Warranted object codes. The table provides data for the 20 Day Weekly by Object and 20 Day Weekly by Department reports.

Input Table: Main Database
Query Type: Make-Table Query

Field Name	*Criteria*	*Description*
Budget_Fiscal_Year		
Document_Code		
Document_Department		
Document_Organization		
Document_Number		
Document_Line		
Object	**[Use 20 Day Weekly Object codes from OBJ2 or dbo_Object] Example:** In ("ee8","h01","h02","h03"…) **-OR-** In ([Weekly Object Code Query]![Object])	Selects 20 Day Weekly object codes, either entered manually, or with a link to an object code query.
Line_Amount		
Date_of_Record		
Payment_Issue_Date		
Actual Days to Pay		

3. Creating the 28 Day Daily Main Table

This table is similar to the 20 Day Weekly table, but due to the large number of object codes it must be broken into two queries. The first query creates the first half of the 28 Day Daily table (object codes D06 to J98), while the second query appends the rest of the object codes to that table (K01 to T19).

A further problem to be considered is that because these transactions will appear on a **Daily Warrant**, they are affected by days in which no cycle is processed (i.e. weekends and holidays). Any transaction whose **Date of Record** falls on a Friday, Saturday, or Sunday may legitimately take as many as 31 days to pay, since no Daily Warrant runs on a weekend. Therefore, it becomes useful to add another field that contains the **Day of Week** that the **Date of Record** occurred on. Such a field may be created by using the function **weekday**, which simply converts a date into a number from 1 to 7 representing the day of the week.

I. Input Table: Main Database
Query Type: Make-Table Query

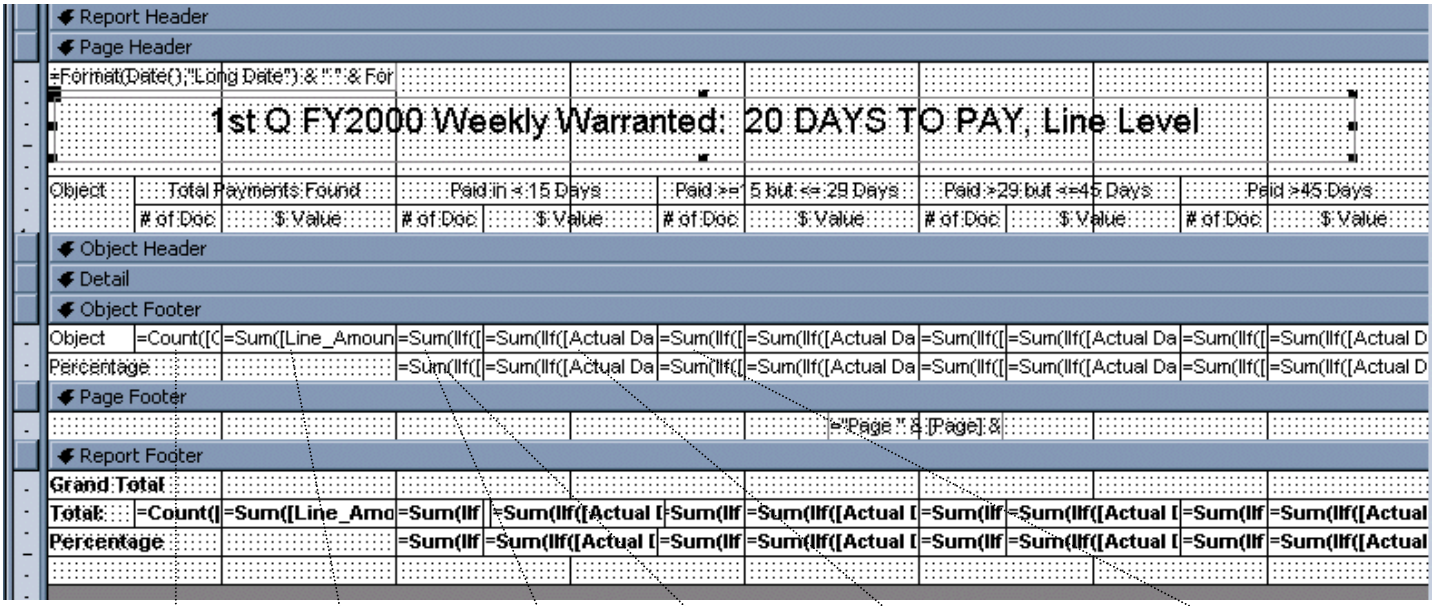
Field Name	*Criteria*	*Description*
Budget_Fiscal_Year		
Document_Code		
Document_Department		
Document_Organization		
Document_Number		
Document_Line		
Object	**[Use 28 Day Daily Object codes from OBJ2 or dbo_Object] Example:** In ("L01","L03","L04","L06","D06"…) **-OR-** In ([Daily Object Code List]![Object])	Selects 28 Day Daily object codes
Line_Amount		
Date_of_Record		
Payment_Issue_Date		
Actual Days to Pay		
Day of Week: weekday (Date_of_Record)		Creates field "Day of Week", which records the day (Mon, Tues, etc) on which a PV was created

Creating the Days to Pay Reports

Since we are more interested in the totals for a particular department or object code, our reports make frequent use of the **=Sum** and **=Count** features of Access reporting. Examples of the formulae used in creating our reports are shown below; more information regarding both the creation of—and the use of expressions within—reports is available in the Access manual.

The main formula used in the Days to Pay reports is based on the SUM and IIF (similar to the IF command in Excel) functions. The structure of the formula is as follows: **SUM(IFF({expression to test}, {value if true}, {value if false}))**. Essentially, the formula is used to calculate whether the value contained in Actual Days to Pay for a given transaction meets the criteria for the column in which it appears (e.g. greater than 15 days *and* less than or equal to 29 days). If it is meets the criteria, the value contained in {value if true} is used. If not, the {value if false} is used instead.

To demonstrate this process, the formula for counting the number of transactions that were paid between 15 and 29 days is as follows: =Sum(IIf([Actual Days to Pay]>=15 And [Actual Days to Pay]<=29,1,0)). This formula can be explained as "if the actual days to pay is 15 or greater *and* is also 29 or less, add 1 to the count; if it is not, add 0".



=Count([Budget_Fiscal_Year])

=Sum([Line_Amount])

=Sum(IIf([Actual Days to Pay]<15,1,0))

=Sum(IIf([Actual Days to Pay]<15,1,0))/Count([Budget_Fiscal_Year])

=Sum(IIf([Actual Days to Pay]<15,[Line_Amount],0))

=Sum(IIf([Actual Days to Pay]>=15 And [Actual Days to Pay]<=29,1,0))

Modifying the Days to Pay Reports (Weekends and Holidays)

Further modifications, such as the one required for transactions entered on Friday, Saturday and Sunday, involve adding additional SUM/IIF statements. These appear in the same format and are simply added after the end of the first statement, preceded by a plus or minus sign.

Wednesday, August25, 1999 11:17:25

MONTH OF JULY FY 99 AND 2000 DAILY DISBURSED: 28 DAYS TO PAY (Adjusted for Fri., Sat., Sun. & July 4th Holiday)

Object	Total Payments Found		Paid In < 11 Days		Paid >=11 but <= 26 Days		Paid >26 but <=29 Days, Plus Fri, Sat, Sun 30 31		Paid >29 but <=45 Days Minus Fri, Sat, Sun 30 31		Paid > 45 Days	
	# of Doc	$ Value	# of Doc	$ Value	# of Doc	$ Value	# of Doc	$ Value	# of Doc	$ Value	# of Doc	$ Value
R11	416	$11,463,196.34	9	$180,137.61	200	$1,899,066.61	141	$9,122,919.62	39	$198,712.41	27	$52,360.09
R14	113	$535,678.53	2	$7,616.66	0	$0.00	107	$412,819.00	0	$0.00	4	$115,242.87
R15	2988	$36,319,772.56	483	$15,080,996.41	1440	$5,032,843.32	162	$2,793,745.66	721	$10,464,094.12	182	$1,948,093.05
R18	221	$195,736.50	5	$2,337.48	2	$7,771.49	197	$185,280.67	11	$23,891.07	6	$6,455.79
R21	456	$632,390.13	109	$68,117.13	24	$3,026.24	222	$506,306.14	25	$2,704.52	76	$52,236.10
R22	1000	$883,336.57	933	$105,200.36	30	$8,560.72	4	$600,703.66	12	$103,680.00	21	$164,191.83
R23	17	$17,167.53	2	$2,422.53	1	$300.00	14	$14,445.00	0	$0.00	0	$0.00
R25	20	$29,977.69	0	$0.00	0	$0.00	20	$29,977.69	0	$0.00	0	$0.00
R26	151	$41,723,356.00	151	$41,723,356.00	0	$0.00	0	$0.00	0	$0.00	0	$0.00
S09	26	$444,914.58	26	$444,914.58	0	$0.00	0	$0.00	0	$0.00	0	$0.00
T01	2	$266,000.00	0	$0.00	2	$266,000.00	0	$0.00	0	$0.00	0	$0.00
T03	1	$199,295.18	1	$199,295.18	0	$0.00	0	$0.00	0	$0.00	0	$0.00
T04	42	$132,211.63	35	$119,259.13	0	$0.00	7	$12,952.50	0	$0.00	0	$0.00
T10	2	$190.73	0	$0.00	0	$0.00	2	$190.73	0	$0.00	0	$0.00
Grand Total												
Total:	93088	$440,316,990.12	11448	$215,764,616.82	6672	$45,724,276.31	97390	$105,340,645.38	10569	$52,471,570.87	7009	$21,015,880.74
Percentage			12.30%	49.00%	7.17%	10.38%	61.65%	23.92%	11.35%	11.92%	7.53%	4.77%

Standard Formula

=Sum(IIf([Actual Days to Pay]=>11 And [Actual Days to Pay]<=26,1,0))

Modified Formula (30, 31)

=Sum(IIf([Actual Days to Pay]>26 And [Actual Days to Pay]<=29,1,0))+Sum(IIf([Actual Days to Pay] In (30,31) And [Day of Week] In (1,6,7),1,0))

Formula now accounts for PVs processed on weekends, and allows these transactions two extra days (since payments are only made Monday through Friday).

Adjusting for Holidays (28 Days Daily Only)

In the above example, an adjustment was made to account for the July 4th holiday (a Monday in this case). The third part of the equation—**Sum(IIf([Actual Days to Pay] In (30,31,32) And [Payment_Issue_Date]=#7/7/99#,1,0))**—accomplishes this by adding to the 3rd column all transactions that were paid in 30, 31, or 32 days *and* were paid on Wednesday, July 7th (the day the holiday-effected transactions would have paid). To modify for other holidays on a monthly or quarterly basis involves changing this date to reflect the holiday in that month. In cases where there is more than one holiday in a month, an **In** statement should be used instead of the = (e.g. **[Payment_Issue_Date] In (#7/7/99#, #7/27/99#)**).

Final Formula: =Sum(IIf([Actual Days to Pay]>26 And [Actual Days to Pay]<=29,1,0))+Sum(IIf([Actual Days to Pay] In (30,31) And [Day of Week] In (1,6,7),1,0))+Sum(IIf([Actual Days to Pay] In (30,31,32) And [Payment_Issue_Date]=#7/7/99#,1,0))

Other Useful Information

Analyzing Document Days to Pay via MS Excel

Once the tables have been created in Access, it may be desirable to print/view the documents in Excel. This is achieved by selecting the table in Access, and choosing the menu item Tools—Office Links—Analyze It with MS Excel. This will automatically convert the table into a spreadsheet, and open it in Excel. If this feature is unavailable, it is possible to manually export a table via the Save As/Export command (under File).

Rolling Tables into Document Level

All transaction records received from the Warehouse are separated into their different lines. One document may have several different lines—each one a different object code—and each line constitutes its own record in the database. While this provides accurate statistics on use of object codes and the total dollar amounts for the lines, because each line is its own record there are discrepancies in the document counts (a four line document is treated as four separate documents). When our document statistics are created, therefore, it is necessary to "roll" the records into document level—to make each record a *unique* document containing the dollar total of all the corresponding lines of that document.

Producing the new Document Level tables involves running an additional query on each of the reporting tables. These queries are set up as follows (20 Days Weekly is used as an example; 28 Days Daily and N Construction queries are similar):

Input Table: 20 Day Weekly Main Table (example)
Query Type: Make-Table Query

Field Name	*Totals/Grouping*	*Description*
Budget_Fiscal_Year	Group By	
Document_Code	Group By	
Document_Department	Group By	
Document_Organization	Group By	
Document_Number	Group By	
Line_Amount	**Sum**	Totals the Line Amount of each document
Date_of_Record	Group By	
Payment_Issue_Date	Group By	
Actual Days to Pay	Group By	

Two fields, **Document_Line** and **Object**, have been left out of this query. Removing these fields ensures that each document appears only once, and that the Line Amounts for all the different lines are totaled.

It should also be noted that in tables created by such queries, the fields that have been totaled or counted receive name changes (e.g. **Line_Amount** becomes **SumOfLine_Amount**). If the fields in the reports do not reflect the name change, Access will return an error when the report is previewed.

Creating the NN Subsidiary Table (Pilot Construction Departments only)

As part of the Commonwealth's construction reform initiative, an additional set of object codes to be analyzed for Days to Pay statistics were a selection of NN Subsidiary codes relating to payments for construction. The specific object codes examined are N15 to N99, with the exceptions of N19, N95, N96 and N99. This table contains all documents that use the aforementioned objects, and provides the data for the NN Subsidiary Days to Pay reports.

Input Table: Main Database
Query Type: Make-Table Query

Field Name	*Criteria*	*Description*
Budget_Fiscal_Year		
Department		
Document_Code		
Document_Department		
Document_Organization		
Document_Number		
Document_Line		
Object	Between "N15" And "N99" And <>N19 And <>N95 And <>N96 And <>N98	Selects desired N Construction object codes
Line_Amount		
Date_of_Record		
Payment_Issue_Date		
Actual Days to Pay		